FIG. 1

Search Criteria

Fill in at least one field. Fill more to narrow your search.
Need high speed? Try Fast Search.

Description:	Stove	202
Manufacturer:	Sears	204
Price:	$500 ▽	206

200

| Search Now | Reset |

208

FIG. 2

300

302 Select	**308** T1.x, T2.y
304 From	**310** T1, T2
306 Where	**312** ————

FIG. 3

Manufacturer

General Electric

Sears

404

Ranges
Stoves
Vacuum

402

Product

Sears   G.E.   Kenmore

Product	Sears	G.E.	Kenmore	
Stove				
Hood				

406

G.E.

400

FIG. 4

```
┌─────────────────┐
│   Initialize    │──── 502
│    System       │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│Initialize Query and│──── 504
│Build Select Clause│
└─────────────────┘
         │
         ▼
┌─────────────────┐
│   Build Where   │──── 506
│     Clause      │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│   Create From   │──── 508
│     Clause      │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  Order by Clause │──── 510
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  Group by Clause │──── 512
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    Execute      │──── 514
│     Query       │
└─────────────────┘
```
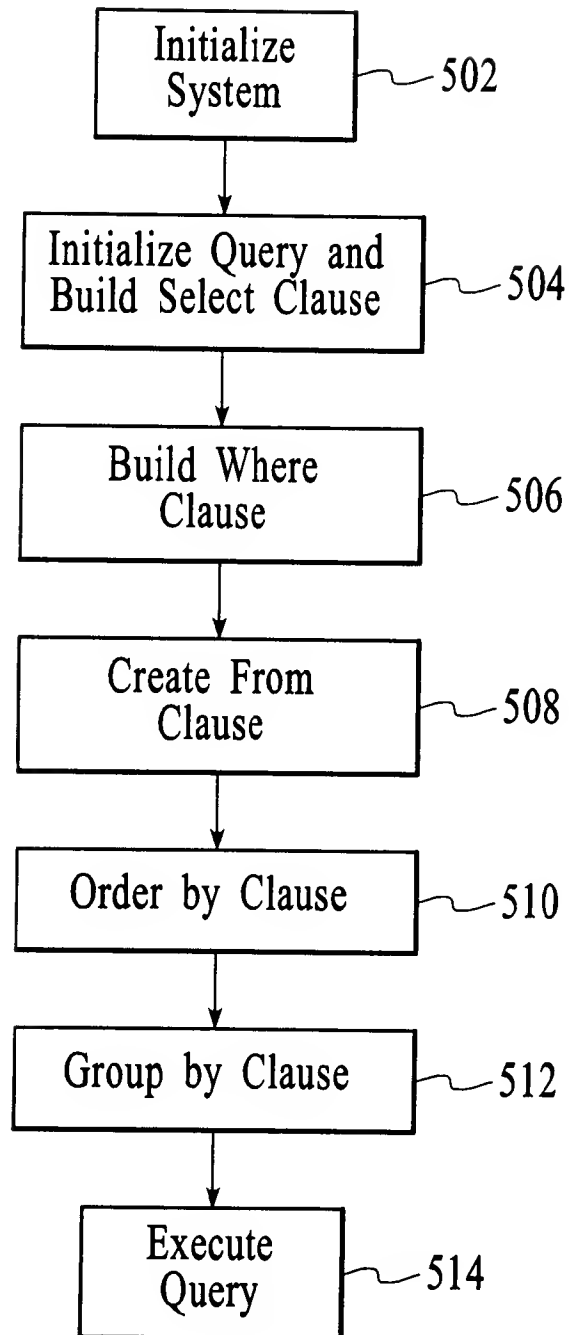
500

FIG. 5

Rose Diagram

FIG. 6

FIG. 7

p2

a4

726 T2.ref ID

710 T1.ref ID

724 =

700

a3

718 =

722 Sears

720 T1.nName

702 and

704 and

p1

a2

708 =

716 red

714 T2.color

a1

706 =

710 T1.ref ID

712 1 2 3
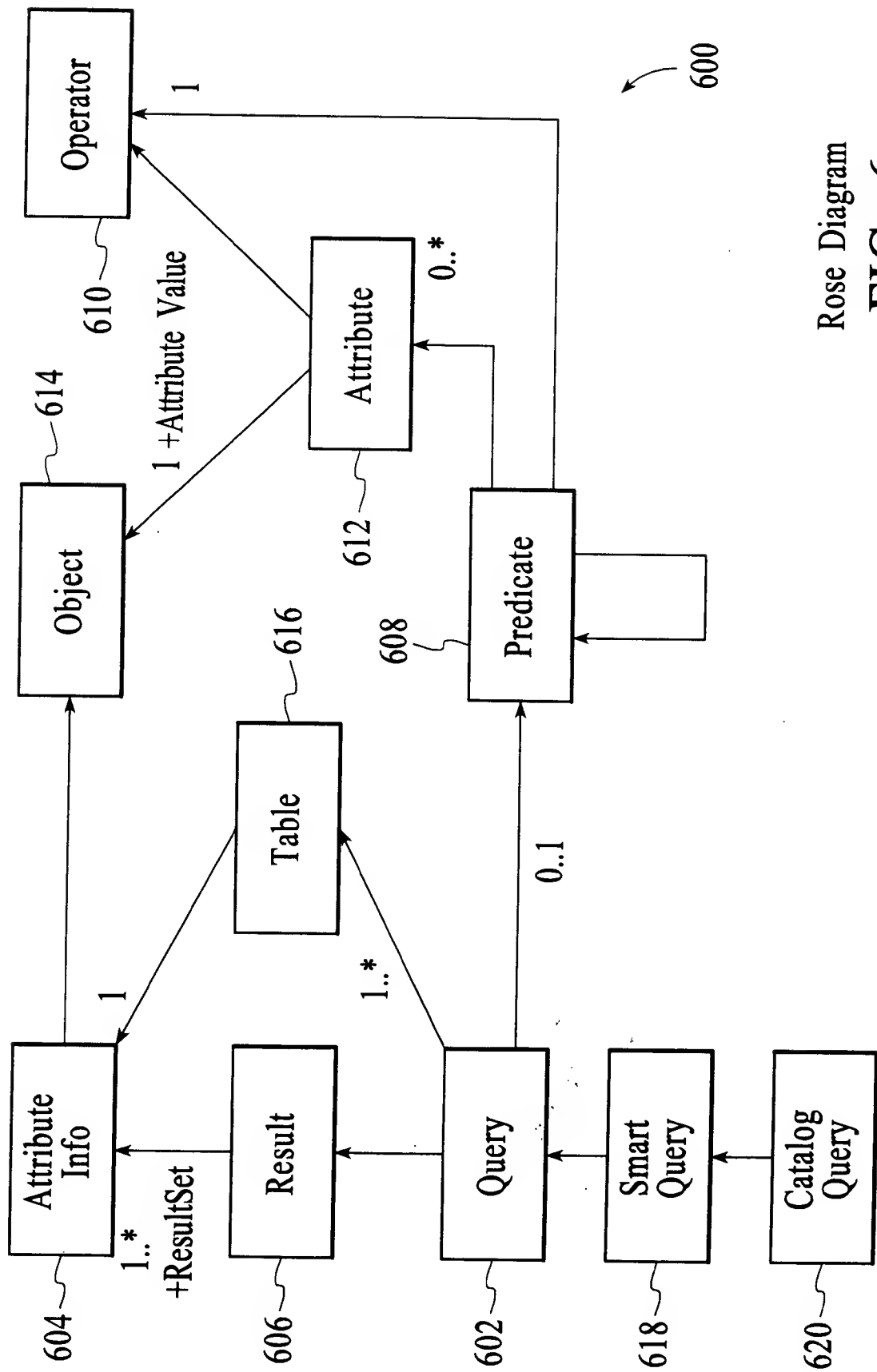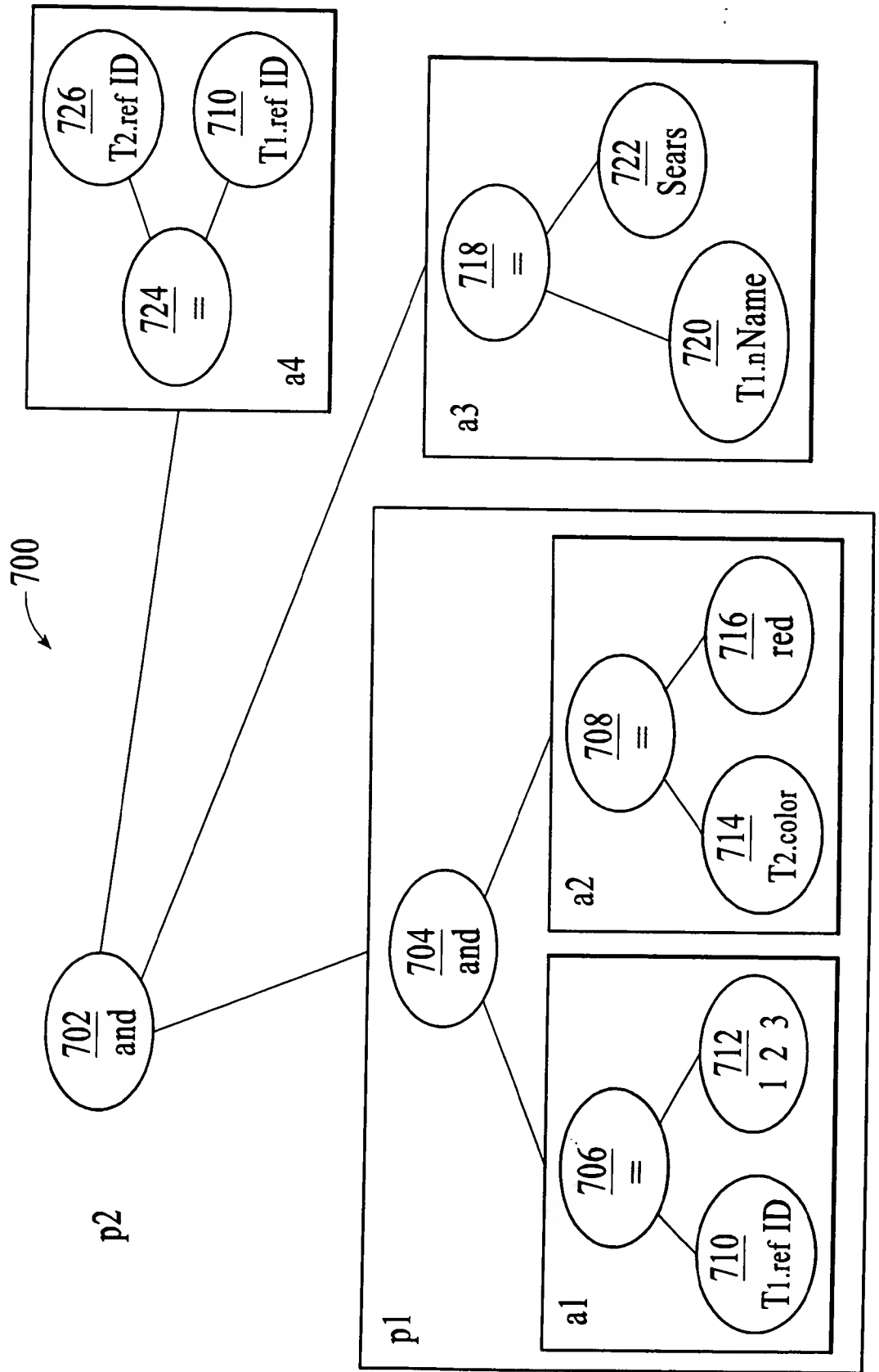
```
        ┌  // construct the simple search conditions
        │  Attribute attr1 = new Attribute (CatRefldAttributeInfo, Operator.equal, "123");
  802  ╱   Attribute attr2 = new Attribute (ColourttributeInfo, Operator.equal, "red");
        ╲  Attribute attr3 = new Attribute (ManufactureAttributeInfo, Operator.equal, "Sears");
        │  Attribute attr4 = new
        └  Attribute (CatRefldAttributeInfo,Operator,equal,DescRefldAttributeInfo,);

        ┌  // compose composite search conditions
  804  ╱   Predicate p1 = new Predicate (Operator.and, {attr1, attr2} );
        ╲  Predicate p2 = new Predicate (Operator.and, {p1, attr3, attr4} );

        ┌  // execute the query
        │  Query q = new Query ( )
  806  ╱   q.setResultSet ({ CatRefldAttributeInfo, ...}) // result set contains catalog entryld
        └  q.setPredicate (p2);
  808 ⌐  result = q.execute ( )
```

<u>800</u>

FIG. 8

```
public void MCQuery( )    throws Exception {

Debug.setLocalTest(true);

System.out.println(" **************** Merchant Centre ****************** ");
```

900 ‿ CatalogQuery MCQuery = new CatalogQuery( );   ‿ 901

```
      // Result set
      MCQuery.addResultSetInfo(new Result(CatEntryIdentifierAttributeInfo.getSingleton( )( )));;
      MCQuery.addResultSetInfo(new Result(StoreInvQuantityAttributeInfo.getSingleton( )( )));;
902 { MCQuery.addResultSetInfo(new Result(CatEntDescShortDescAttributeInfo.getSingleton( )( )));;
      MCQuery.addResultSetInfo(new Result(CatEntDescNameAttributeInfo.getSingleton( )( )));;
      MCQuery.addResultSetInfo(new Result(CatEntryTypeAttributeInfo.getSingleton( )( )));;
      MCQuery.setDistinctQualifier(true);    ‿ 904
```

FIG. 9A(i)

```
// Predicate set
// Part I
Predicate p11 = new Predicate ( );    ～908a
p11.setOperator (Operator.or);
Attribute a111 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leftlike,
"CATEGORY X");
a111.setUppercaseQualifier(true);
p11.addOperand (a111);
Attribute a112 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leftlike,
"CATEGORY10"
a112.setUppercaseQualifier(true);
p11.addOperand (a112);
```
910a

```
Predicate p12 = new Predicate ( );    ～908a
p12.setOperator (Operator.and);
p12.addOperand (new Attribute (ListPriceAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p12.addOperand (new Attribute (StoreInvQuantityAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p12.addOperand (new Attribute (InventoryQuantityMeasureAttributeInfo.getSingleton( ), Operator.isnull));
p12.addOperand (p11);
```
910a

```
Predicate p13 = new Predicate ( );    ～908a
p13.setOperator (Operator.and);
p13.addOperand (new Attribute (ListPriceAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p13.addOperand (new Attribute (StoreInvQuantityAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p13.addOperand (new Attribute (InventoryQuantityMeasureAttributeInfo.getSingleton( ), Operator.isnull));
Attribute a13 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leflike,
"CATEGORY5");
a13.setUppercaseQualifier(true);
p13.addOperand (a13);
```
910a

```
Predicate p14 = new Predicate ( );    ～908a
p14.setOperator (Operator.or);
p14.addOperand (p12);
p14.addOperand (p13);
```
910a

FIG. 9A(ii)

```
// Part II
Predicate p2 = new Predicate ( );        ~908b
p21.setOperator (Operator.or);
Attribute a211 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leftlike,
"CATEGORY Z");
a211.setUppercaseQualifier(true);
p21.addOperand (a211);
Attribute a212 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leftlike,
"CATEGORY9"
a212.setUppercaseQualifier(true);
p21.addOperand (a212);
```

910b

```
Predicate p22 = new Predicate ( );        ~908b
p22.setOperator (Operator.and);
p22.addOperand (new Attribute (ListPriceAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p22.addOperand (new Attribute (StoreInvQuantityAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p22.addOperand (new Attribute (InventoryQuantityMeasureAttributeInfo.getSingleton( ), Operator.isnull));
p22.addOperand (p21);
```

910b

```
Predicate p23 = new Predicate ( );        ~908b
p23.setOperator (Operator.and);
p23.addOperand (new Attribute (ListPriceAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p23.addOperand (new Attribute (StoreInvQuantityAttributeInfo.getSingleton( ), Operator.gt, "0.0"));
p23.addOperand (new Attribute (InventoryQuantityMeasureAttributeInfo.getSingleton( ), Operator.isnull));
Attribute a23 = new Attribute (CatGrpDescNameAttributeInfo.getSingleton( ), Operator.leflike,
"CATEGORY4");
a23.setUppercaseQualifier(true);
p23.addOperand (a23);
```

910b

```
Predicate p24 = new Predicate ( );        ~908b
p24.setOperator (Operator.or);
p24.addOperand (p22);
p24.addOperand (p23);
p24.setNotQualifier(true);
System.out.println(p24.toString( ));
```

910b

FIG. 9A(iii)

```
// Part IV -- Join
Predicate p4 = new Predicate ( );  ～912
p4.setOperator (Operator.and);
p4.addOperand (p14);
p4.addOperand (p24);
p4.addOperand (new Attribute (StoreCEntStoreIdentifierAttributeInfo.getSingleton( ),  ～914
Operator.eq, "2"));
p4.addOperand (new Attribute (UsersIdentifierAttributeInfo.getSingleton( ), Operator.eq,
"1001"));
//p4.addOperand (p33);


MCQuery.setPredicate(p4);  ～916


// Join
System.out.println("Auto Join : ");
MCQuery.printJointRelationships( );


// Resolve source tables
MCQuery.resolveSourceTables( );  ～918


// ORDER, GROUP and HAVING set
MCQuery.addResultOrder(CatEntryIdentifierAttributeInfo.getSingleton( ),
Operator.desc);  ～920

System.out.println("MC Query : ");
System.out.println(MCQuery.toString( ));


com.ibm.commerce.base.objects.Cursor cursor = new
com.ibm.commerce.base.objects.Cursor( );
java.util.Vector v = MCQuery.execute(cursor);  ～922
System.out.println("MC Query first 10 Result: ")
System.out.println(v);
cursor.increment( );
v = MCQuery.execute(cursor);  ～922
System.out.println("MC Query next 10 Result: ");
System.println(v);


}
```

FIG. 9A(iv)

```
public void setPredicate(Predicate aPredicate) throws Exception {

    Predicate additionalP = additionalPredicate( );    ~924
    if (additionalP != null) {

        Predicate p = new Predicate( );
        p.setOperator(Operator.and);
        p.addOperand(aPredicate);
        p.addOperand(additionalP);
        setTableJointPredicate(p);
    }
    else
        setTableJointPredicate(aPredicate);
}
```
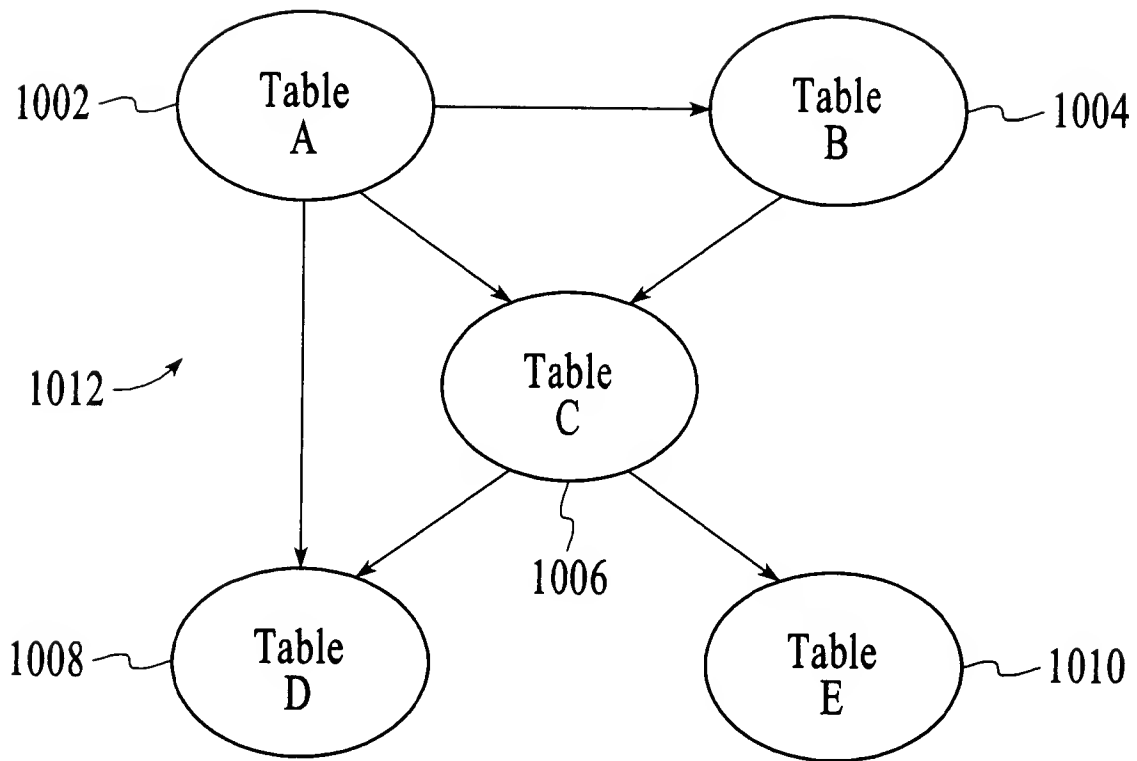
926

```
private void setTablejointPredicate(Predicate aPredicate) throws Exception {

    Predicate jointP = resolveJointPredicate(aPredicate);
    if (jointP != null) {

        Predicate p = new Predicate( );
        p.setOperator(Operator.and);
        p.addOperand(aPredicate);
        p.addOperand(jointP);
        super.setPredicate(p);
    }                              ~930
    else
        super.setPredicate(aPredicate);
}
```
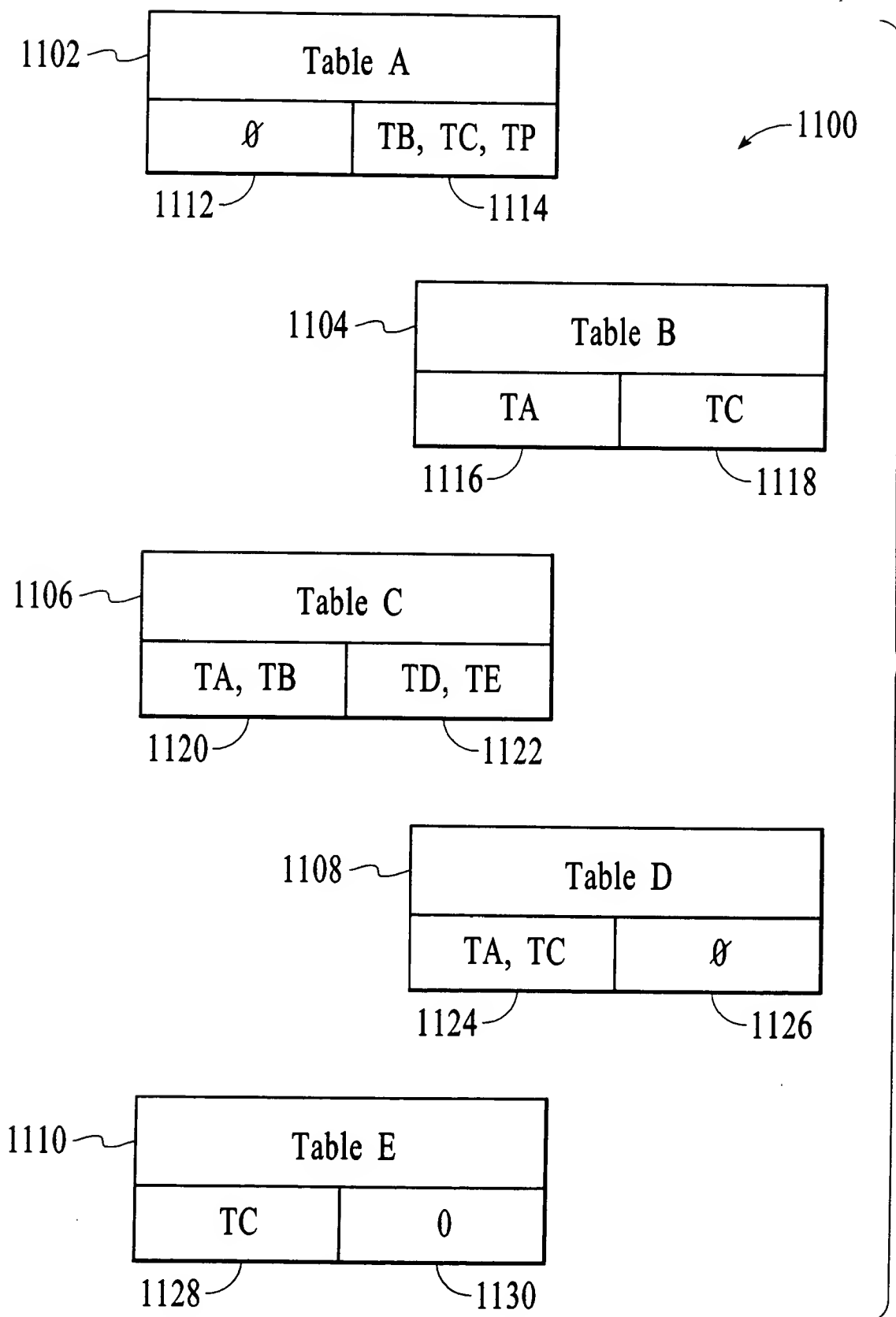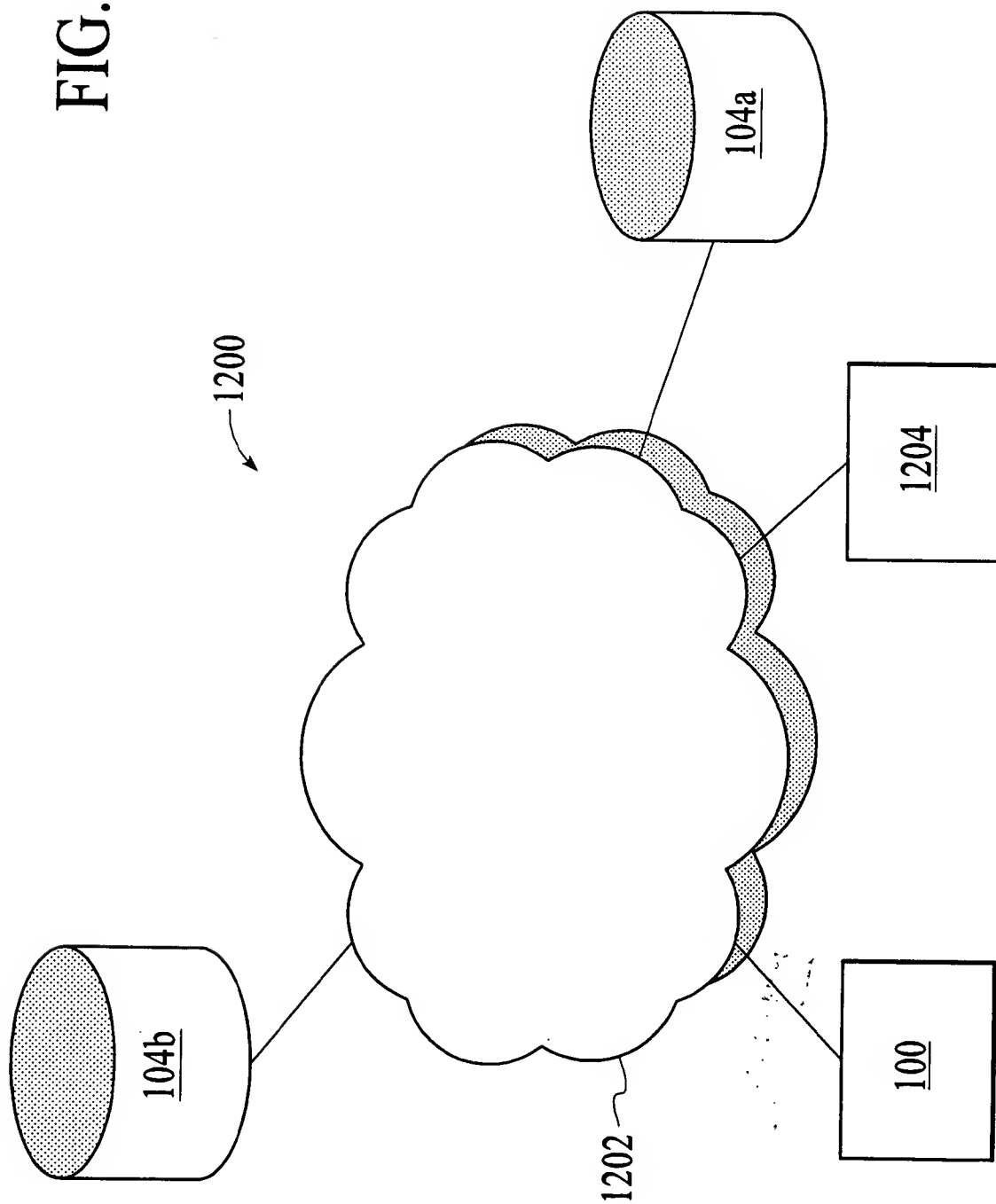
FIG. 9B

1002 — Table A

Table B — 1004

1012 —

Table C

1006

1008 — Table D

Table E — 1010

1000

FIG. 10

1102 —

Table A	
Ø	TB, TC, TP

1112   1114

1100

1104 —

Table B	
TA	TC

1116   1118

1106 —

Table C	
TA, TB	TD, TE

1120   1122

1108 —

Table D	
TA, TC	Ø

1124   1126

1110 —

Table E	
TC	0

1128   1130

FIG. 11

FIG. 12

1200

1202

104a

104b

1204

100